UNIVERSITY OF MARYLAND
ROBERT H. SMITH SCHOOL OF BUSINESS

ASSIGNMENT 3

Your objective is to develop models to predict the outcome variable "BadBuy", which labels whether a car purchased at an auction was a "bad buy" (lemon). Your task is to build a model to guide auto dealerships in their decisions on whether to bid for and purchase a vehicle. You can also apply your learning from this analysis to make more data-informed car-buying decisions!

You will use **carvana.csv** which contains data from 10,062 car auctions as provided by Carvana. Auto dealers purchase used cars at auctions with a plan to sell them to consumers, but sometimes these auctioned vehicles can have severe issues that prevent them from being resold at a profit (hence, lemons). The data contains information about each auctioned vehicle.

Data Dictionary

Variable	Definition
Auction	Auction provider where vehicle was purchased
Age	The years elapsed since the manufacturer's year (how old is the vehicle)
Make	Vehicle manufacturer
Color	Vehicle color
WheelType	Vehicle wheel type description (Alloy, Covers)
Odo	Vehicle odometer reading
Size	Size category of the vehicle (Compact, SUV, etc.)
MMRAauction	Auction price for this vehicle (in average condition) at the time of purchase
MMRAretail	Retail price for this vehicle (in average condition) at the time of purchase
BadBuy	Whether the vehicle is a bad purchase / lemon ("YES") or a good investment ("NO")

Before you start:

- Load the following libraries in the given order: *tidyverse, tidymodels, plotly, skimr, caret*
- Load the Carvana data and call it *dfc*
- Explore the dataset using `skim()` etc.

Assignment Instructions

There are two main objectives. The first is to predict the variable BadBuy as a function of the other variables. The second is to build alternative models, measure, and improve performance.

1) (~5 points) Data preparation

a) Load the dataset into R and call it *dfc*. Inspect and describe the data.

Ans- Dataset consists of five numerical and five categorical variables. Some categorical variables have a lot of levels. There are 10000+ observations and 10 columns, which is a good distribution for prediction models. The dependent variable - BadBuy has two levels - 0 and 1. Age and Odo seem to have a normal distribution whereas MMRAauction and MMRAretail are right skewed.

b) Set the seed to **52156**. Randomly split the dataset into a training dataset and a test dataset. Use **65%** of the data for training and hold out the remaining **35%** for testing.

2) (~10 points) Exploratory analysis of the *training* data set

a) Construct and report boxplots of the (1) auction prices for the cars, (2) ages of the cars, and (3) odometer of the cars broken out by whether cars are lemons or not. Does it appear that there is a relationship between either of these numerical variables and being a lemon? Describe your observations from the box plots. Please also pay attention to the outliers detected by the box plots and make sense of them.

Ans- Auction Prices - Median auction price for lemons is slightly less as compared to median of good investments. This does not seem to be a huge difference though. However, there are a lot of outliers in the auction price for lemons, indicating many bad buys also get sold at very high prices, further implying it's difficult to guess their true value beforehand.

Median age of lemons is more as compared to good cars, which is expected behaviour. As the age of a car increases, it is usually more used, leading to decrease in performance and increasing chances of becoming a lemon. There is also just one outlier in the good investment category, indicating less variation. Thus, there seems to be a good correlation between age and a car being lemon.

Median is higher for lemons in case of odometer reading, which is expected as the distance covered by a car is more, there are chances of more wear and tear, leading to higher chances of becoming a lemon. However the difference in medians is not too much, indicating this might not be a good indicator. There are a lot of outliers in case of

lemons, indicating there are a lot of lemons that have low odometer readings. Thus reinforcing the fact that this factor alone might not be a good indicator.

b) Construct and report a table for the count of good cars and lemons broken up by Size (i.e., How many vehicles of each size are lemons?).
Hint: Remember `tally()`? That's one way to do it. You may want to think more systematically and use a combination of summarize(), length(), mutate(), arrange()

i) Which size of vehicle contributes the most to the number of lemons? (That is, which vehicle size has the highest *percentage* of the total lemons?)

Ans - Medium size has the highest percentage of total lemons.

ii) Because the vehicles of the size you identified in (i) contribute so much to the number of lemons, would you suggest the auto dealership stop purchasing vehicles of that size? Why or why not?

Ans - No, it is not suggested to stop purchasing medium sized vehicles, as the high value of good investments of medium sized cars indicates that this is simply the most popular category of cars sold. The correlation between a car being lemon because of being medium sized can't be inferred from this.

3) (~20 points) Run a linear probability model to predict a lemon using all other variables.

a) Compute and report the RMSE using your model for both the training and the test data sets. Use the predicted values from the regression equation. **Do not** do any classifications yet.

Ans - RMSE for Training Set = 0.4479, RMSE for Test Set = 0.4528

b) For which dataset is the error smaller? Does this surprise you? Why or why not?

Ans - Error is smaller for training dataset. This is not surprising, as the model was trained using the training dataset and usually test set has a higher error as compared to training set due to this. However, since we used all the independent variables without checking which variables actually influence the dependent variable, this might indicate the model has been overfit, and/or captures noise instead of the actual trends in dataset. Thus indicating a high variance, low bias model.

c) Use a cutoff of 0.5 and do the classification in the test data. Compute and report the confusion matrix (recall to convert BadBuy into a factor for the confusion matrix).

i) Which type of errors (false positives and false negatives) occur more here?

Ans - False negative error is higher than false positive error.

ii) For this problem, do you think a false positive or a false negative is a more serious error? Based on your answer, which metric makes a better objective?

Ans - Since this model is primarily for buyers, false negative error is more serious. If a car is actually a lemon and is predicted as a good investment, dealers might buy that car and would have to endure losses.
Sensitivity is the best metric here.

d) What is the testing accuracy of your model? Based on accuracy, does the model perform better than using a random classifier (i.e., the baseline accuracy)?
Hint 1: Calculate manually if you like, or use the `confusionMatrix()` function.
Hint 2: The baseline accuracy is the accuracy you would achieve if you classified every single class as a member of the most frequent class in the actual test dataset.

Ans - Accuracy is 67.31%.
Baseline accuracy = 50.61%
This indicates our model is better than using a random classifier.

e) Compute and report the predicted "probability" that the following car is a lemon:
Auction="ADESA" Age=1 Make="HONDA" Color="SILVER"
WheelType="Covers" Odo=10000 Size="LARGE"
MMRAauction=8000 MMRAretail=10000
Does the probability your model calculates make sense? Why or why not?

Ans - Probability = -0.1185. Probability should lie between 0 and 1, thus this probability value doesn't make sense.

4) (~25 points) Run a logistic regression model to predict a lemon using all other variables.
Hint 1: Don't forget to convert your dependent variable BadBuy to a factor in both datasets.
Hint 2: If you haven't yet, switch to using *caret* at this point.

a) Did you receive a rank-deficient fit error? Why do you think so? Figure out the variables causing the problem by running tally() for all your factor variables, and recode them in a way to prevent the error.
Hints: You will need to recode two factor variables:
1. *Color* has two redundant levels that need to be combined.
2. Create a new category for *Make*, call it OTHER, and recode any of the makes with less than 10 observations as OTHER.

Ans - Error is due to redundant/irrelevant levels. This error is usually thrown when R considers the model is too complex and has useless features or when the number of ranks is less than the number of rows/columns. Here, upon inspecting Color and Make variables we can see that there are redundant ranks which can be removed.

Make sure to make the changes in the full dataset, convert BadBuy to a factor, repeat the process of setting the seed to 52156 and splitting the data.
Run your logistic regression again to confirm the rank-deficient fit error is gone.

b) What is the coefficient for Age? Provide an exact numerical interpretation of this coefficient.

Ans - Coefficient is 0.2785. exp(0.2785) = 1.32.
An increase in age by 1 unit is associated with an increase in the odds of a car being a lemon by a factor of 1.32, when everything else remains constant.

c) What is the coefficient for SizeVAN? Provide an exact numerical interpretation of this coefficient.

Ans - Coefficient is -0.5982. Odds ratio = exp(-0.5982) = 0.5498.
The odds of a van being a lemon are 0.5498 times the odds of a vehicle of size compact being a lemon, provided everything else remains constant.

d) Use a cutoff of 0.5 and do the classification in the test data. Compute and report the confusion matrix for your test data predictions.

Ans -

predictedClass	BadBuy/True 0	1
0	1341	721
1	441	1018

e) Compute and report the predicted probability using your logistic model for the same car from 3(e). What does the resulting value tell you about this particular car now? Does the result make more sense than the result in Question 3(e)? Why or why not?

Ans - logit = -3.2724. Probability = 0.036. Since the probability is 0.036, it indicates the car is classified as not a lemon, since it is very close to 0 and less than the cutoff of 0.5. This makes sense and helps us predict the class clearly, unlike in 3e.

Pro tip: Pipe a confusion matrix (from any model) into tidy() and see what happens!

(5) (~40 points) Explore alternative classification methods to improve your predictions.

- In the models below, use a 10-fold cross validation to make the results consistent across.
- Use the same training and test data you created and used after recoding the data in Q4.
- Make all comparisons to the logistic model you have run in Q4 after recoding the data.

a) Set the seed to **123** and run a linear discriminant analysis (LDA) using all variables.

i) Compute the confusion matrix and performance measures for the test data, and compare them **with the logistic regression results**. Discuss your findings.

Ans - Logistic regression model has an accuracy of 67% and sensitivity of 58.5%. LDA has accuracy of 67.23% and sensitivity of 56.9%.

Since logistic model has almost similar accuracy and higher sensitivity, it is preferred as sensitivity is an important metric here.

b) Set the seed to **123** and run a kNN model using all variables.
   i) Create a plot of the k vs. cross-validation accuracy.
   ii) What is the optimal k? What else do you infer from the plot?
   Hint: To inspect the details of any model, you will need to train the model and store it before piping it into predict(). See the GitHub repository for guidance.
   Ans - Optimal k i 19.
   As the value of k increases, accuracy goes on increasing till 19, after which it drops.
   iii) Compute the confusion matrix and performance measures for the test data, and compare them **with the logistic regression and LDA model** results. Discuss your findings.
   Ans - Accuracy of kNN is 62.85% and sensitivity is 55%, both are lower than both Logistic and LDA models. Thus, logistic still remains the best model.

c) Set the seed to **123** and build a lasso model using all variables.
   i) Set the seed to **123** and run a Lasso model using all variables. Report the table of variable importance in a tibble format and share your observations.
   Hint: See the Github repo for help. Use a 100-point grid between 10^{-5} and 10^{2}
   Ans - All the top variables of importance seem to be categorical ones, and lasso has dropped a few levels within them, indicating this might be a very useful result.

   ii) Report the plot of variable importance for the 25 most important variables.

   iii) What is the optimum lambda selected by the model? What does it mean that the algorithm chooses this particular lambda value?
   Ans - `0.0003053856`
   Lasso tries to find the lamda values so that variables that have maximum effect on the dependent variables can be chosen and other discarded. Algorithm chose this lambda because it gives the minimum error rate on validation data.

   iv) Compute the confusion matrix and performance measures for the test data, and compare them **with the logistic regression, LDA, and kNN model** results. Discuss your findings.

Ans - Lasso has accuracy of 66.94% and sensitivity of 58.5%. COmparing the metrics, logistic is still the better model.

d) Set the seed to **123** and build a (I) ridge and (II) elastic net[1] model using all variables.
   i) Compute the confusion matrix and performance measures for the test data, and compare them **only with the lasso model** results. Discuss your findings. **Hint:** Use the same grid for lambda. Notice the different optimum value!

Ans - Ridge has accuracy of 66.1% and sensitivity of 59.8%.
Elastic net has accuracy of 66.8% and sensitivity of 58.4%.
Compared to lasso mode, both of them show better accuracy and sensitivity values.
Better performance of ridge and elastic net indicates presence of correlated variables. Lasso tries to minimize the individual coefficients and Ridge focuses on creating a robust model.

e) Set the seed to **123** and run a quadratic discriminant analysis (QDA) with all variables
   i) Have you received an error? What do you think the error you received means? Do some research and explain what you think it is about.

Ans - The rank deficit error indicates collinearity between variables. QDA doesn't assume equality of covariance among predictor variables and checks for the same. When there is collinearity and covariance matrices can't be inverted to obtain the estimates in control, it throws this error.

   ii) Why is the rank deficiency a problem for QDA, but not for LDA?

Ans - LDA assumes equality of covariance and doesn't check for it. QDA doesn't assume equality of covariance among predictor variables and checks for the same and checks for it.

   iii) Compute the confusion matrix and performance measures for the test data, and compare them **only with the LDA model** results. Discuss your findings.

Ans - QDA has accuracy of 63.87% and LDA has accuracy of 67.23%, indicating LDA is better. This could be because there could be identical covariance between variables, and a linear boundary defines classes better rather than a quadratic boundary.

f) Among all the models you have studied, which model do you think is better for the given business case/problem? Discuss why you think it is better than the others. Also report the ROC curves for the models you have developed on the same chart.

Ans - Logistic model seems to be best for this case, as it gives highest accuracy and sensitivity values. LDA can also be used as it also gives good accuracy and sensitivity measures.

[1] Naive elastic net. Feel free to run a grid search but be careful not to hit the limits of your computational power!

ROC curves show that most models have similar performance with respect to sensitivity and specificity, except QDA and KNN, which do not perform well for this dataset.

Bonus question: You may have noticed that lasso drops certain levels of Make and Color such as “Brown”, keeping the other levels of the same variable (“Blue” etc.). This may not be helpful, so you may want to use a grouped lasso. Set the seed to 123 and try grouped lasso with the lambda values 50 and 100. Do the results make more sense now? Why or why not?
Ans - Dropping certain levels from a single variable doesn’t provide meaningful results since the interpretation of variables depends on the comparison of levels with the control. Adding some levels and ignoring other would lead to the creation of an inaccurate model. Thus, adding group lasso gives a better and more intuitive model. Grouped lasso shows that only certain categorical variables have an effect on predicting the class and some of the categorical variables such as color, size, make have been completely dropped.

Hint: Run a plain lasso again with a lambda value of 0.01 and print the coefficients this time. Compare them with the coefficients from group lasso.